

DAC Characterization

INTRODUCTION TO DIGITAL LOW-LEVEL RADIO
FREQUENCY CONTROLS IN ACCELERATORS

Lab 7
Qiang Du

US PARTICLE ACCELERATOR SCHOOL
JANURARY 23 – 27, 2023

Contents

1	Introduction	2
1.1	Firmware frequency setting	2
1.2	Glossary	2
1.3	Hardware setup	3
1.4	Firmware and software setup	3
2	Exercises	3
2.1	Generate a Sine wave at 20 MHz	3
2.2	Check DAC waveform on the oscilloscope	5
2.3	Check DAC frequency response on spectrum analyzer	6
2.4	Change DAC output frequency	7

1 Introduction

In this lab we will learn how to use a digital LLRF system as a signal generator by directly controlling the DDS and DAC. We will also learn how to characterize the DAC itself, and compare against manufacturer's specifications for testing the system's performance.

1.1 Firmware frequency setting

Frequency	Derivation	Value	Unit
f_MO		480	MHz
f_IF	f_MO/24	20	MHz
f_LO	f_MO - f_IF	460	MHz
f_CLK	f_LO / 4	115	MHz
f_IF / f_CLK		4 / 23	

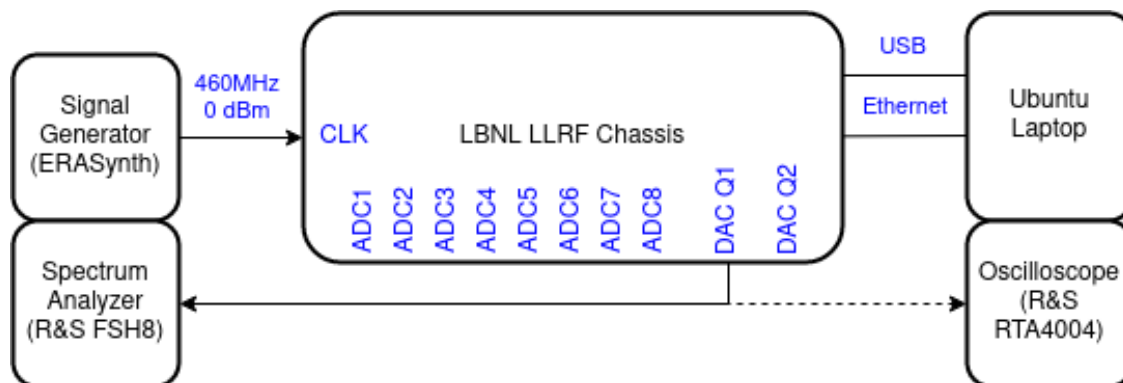
1.2 Glossary

- **f_MO** Master Oscillator frequency, equals to the frequency of operation, or cavity RF frequency.
- **f_IF** [Intermediate Frequency](#). RF signal processing is often performed at a frequency significantly below the frequency of operation. Signal at IF is typically produced by a [superheterodyne receiver](#).
- **f_LO** [Local Oscillator](#) frequency.
- **f_CLK** ADC / DAC sampling clock, which is equal to the clock frequency of DSP logic.

DAC [Digital-to-analog converter](#), an electronics device that converts a sequence of digital codes to corresponding analog voltages or currents. We use Analog Devices [AD9781](#) in [Zest](#) digitizer board in the LLRF system.

DDS [Direct Digital Synthesis](#). A technique for generating arbitrary frequencies and waveforms from a fixed-frequency clock source.

1.3 Hardware setup



1.4 Firmware and software setup

- Configure FPGA chassis using the provided `marble_zest_top_uspas.bit`;
- Start EPICS IOC on the connected laptop computer;

2 Exercises

2.1 Generate a Sine wave at 20 MHz

Connect to DAC Q1 output from the LLRF chassis to the Spectrum Analyzer. Set the output level by adjust the open loop amplitude setpoint:

```
$ caput USPAS:LLRF:Loop:AmpSetp 30000
```

The raw DAC output data is buffered in a memory block, which can be read out. Capture the DAC waveform using the following example. Plot the data with horizontal axis in time and vertical axis in DAC counts.

Reproduce the following example of: * Record DAC1 data; * Calculate its power spectrum;

```
[2]: %matplotlib inline
from matplotlib import pyplot as plt
from scipy import signal
import pandas as pd
import numpy as np
from epics import PV

plt.rcParams['figure.figsize'] = [6, 4]
plt.rcParams['axes.grid'] = True
plt.rcParams['axes.grid.which'] = "both"
plt.rcParams['grid.linewidth'] = 0.5
plt.rcParams['grid.alpha'] = 0.5
plt.rcParams['font.size'] = 8
```

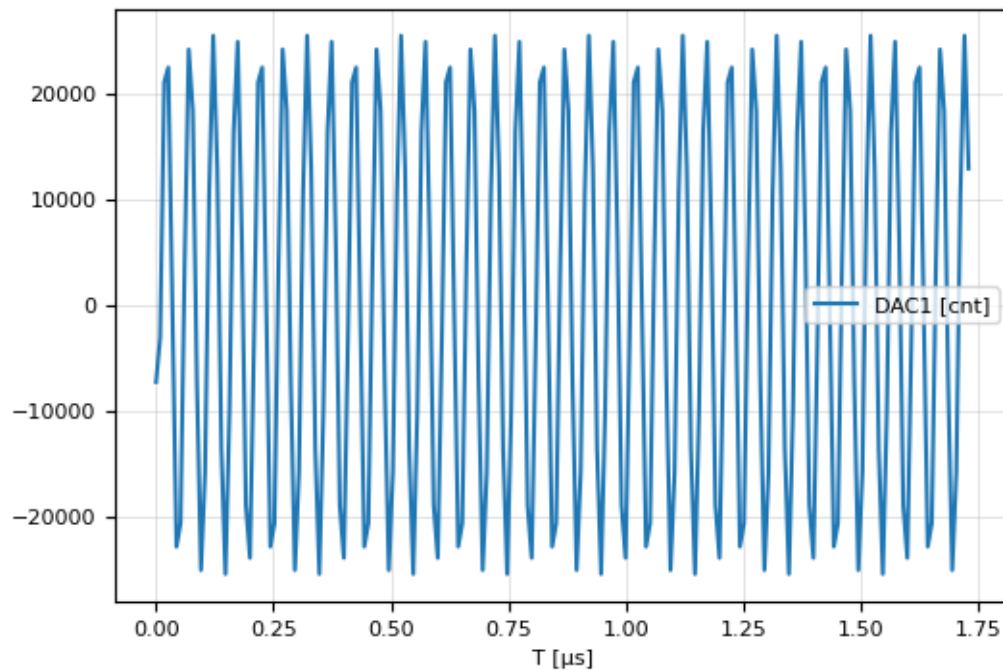
Acquire data of 4096 samples:

```
[3]: pv = PV('USPAS:LLRF:reg_dac0_buf_RBV')
     dac_dat = pv.value
```

Plot time domain data:

```
[5]: fs = 480 * 23 / 24 / 4 # MHz
     t = np.arange(len(dac_dat)) / fs #  $\mu s$ 

     df = pd.DataFrame({
         'T [ $\mu s$ ]: np.arange(len(dac_dat)) / fs,
         'DAC1 [cnt]': dac_dat})
     df[:200].plot(x='T [ $\mu s$ ']);
```



Calculate power spectrum:

```
[16]: fs = 480 * 23 / 24 / 4 # MHz
     fullscale = 1 << (16 - 1) # signed 16 bit format, AD9781
     fsdb = 20 * np.log10(fullscale / np.sqrt(2))
     f, psd = signal.periodogram(
         df['DAC1 [cnt]'], fs, 'flattop', scaling='spectrum')
     psd_dbfs = 10 * np.log10(psd) - fsdb
     df_psd = pd.DataFrame(
```

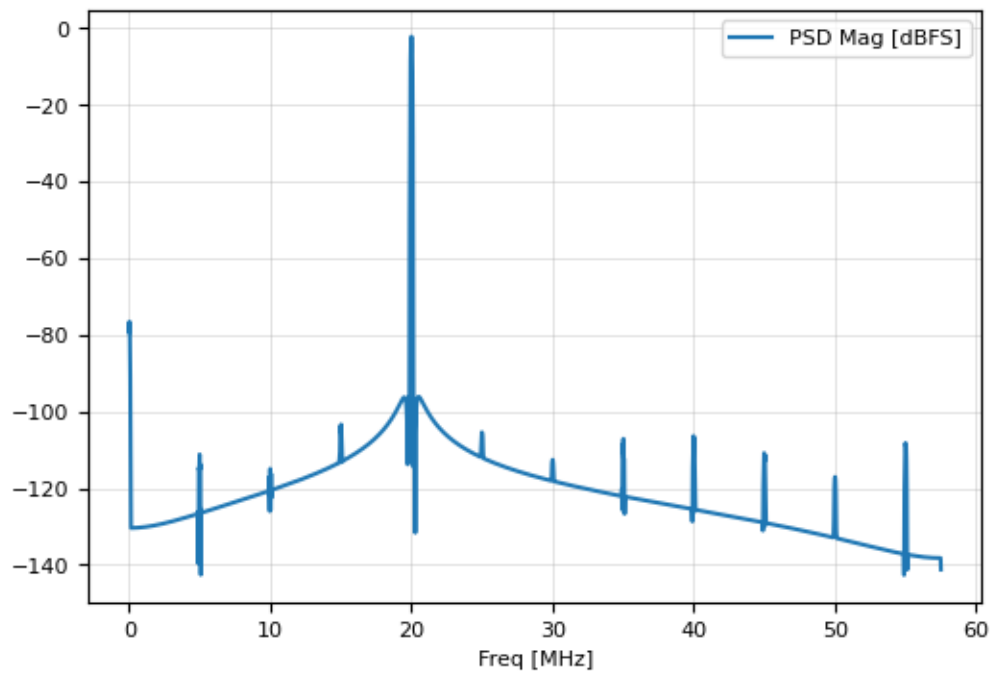
```

    {'Freq [MHz]': f, 'PSD Mag [dBFS]': psd_dbfs})
df_psd.plot(x='Freq [MHz]');
print(f'Peak frequency: {f[np.argmax(psd)]:.3f} MHz')
print(f'Peak magnitude: {psd_dbfs[np.argmax(psd)]:.3f} dBFS')

```

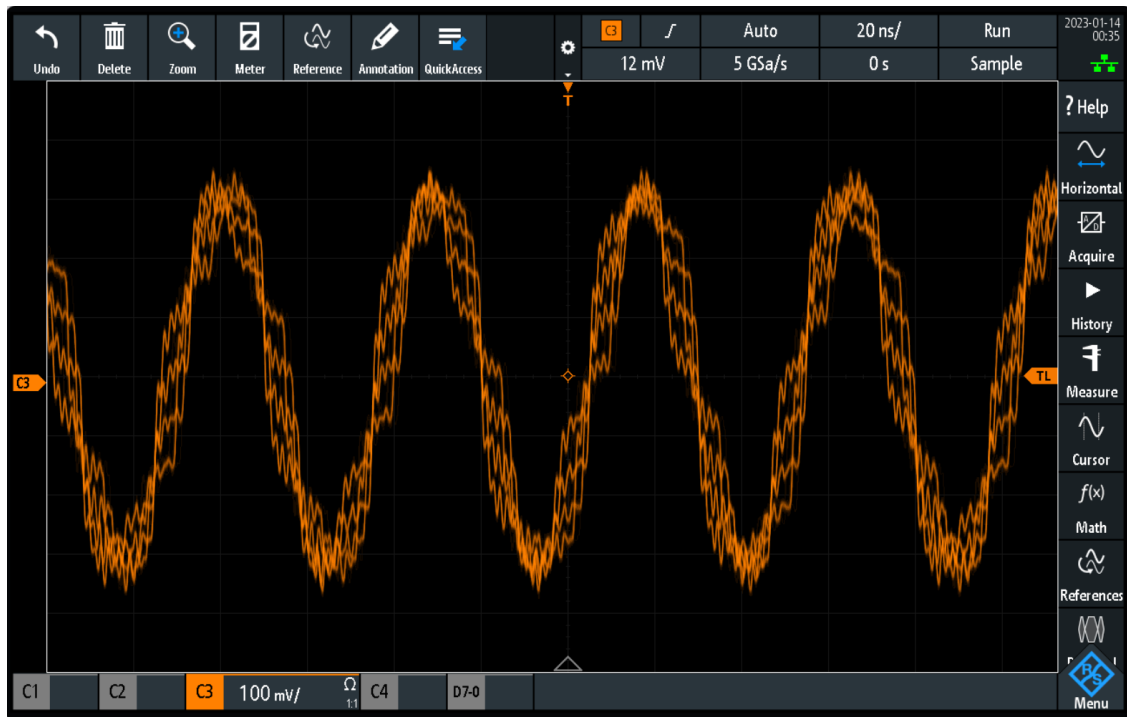
Peak frequency: 19.990 MHz

Peak magnitude: -2.161 dBFS



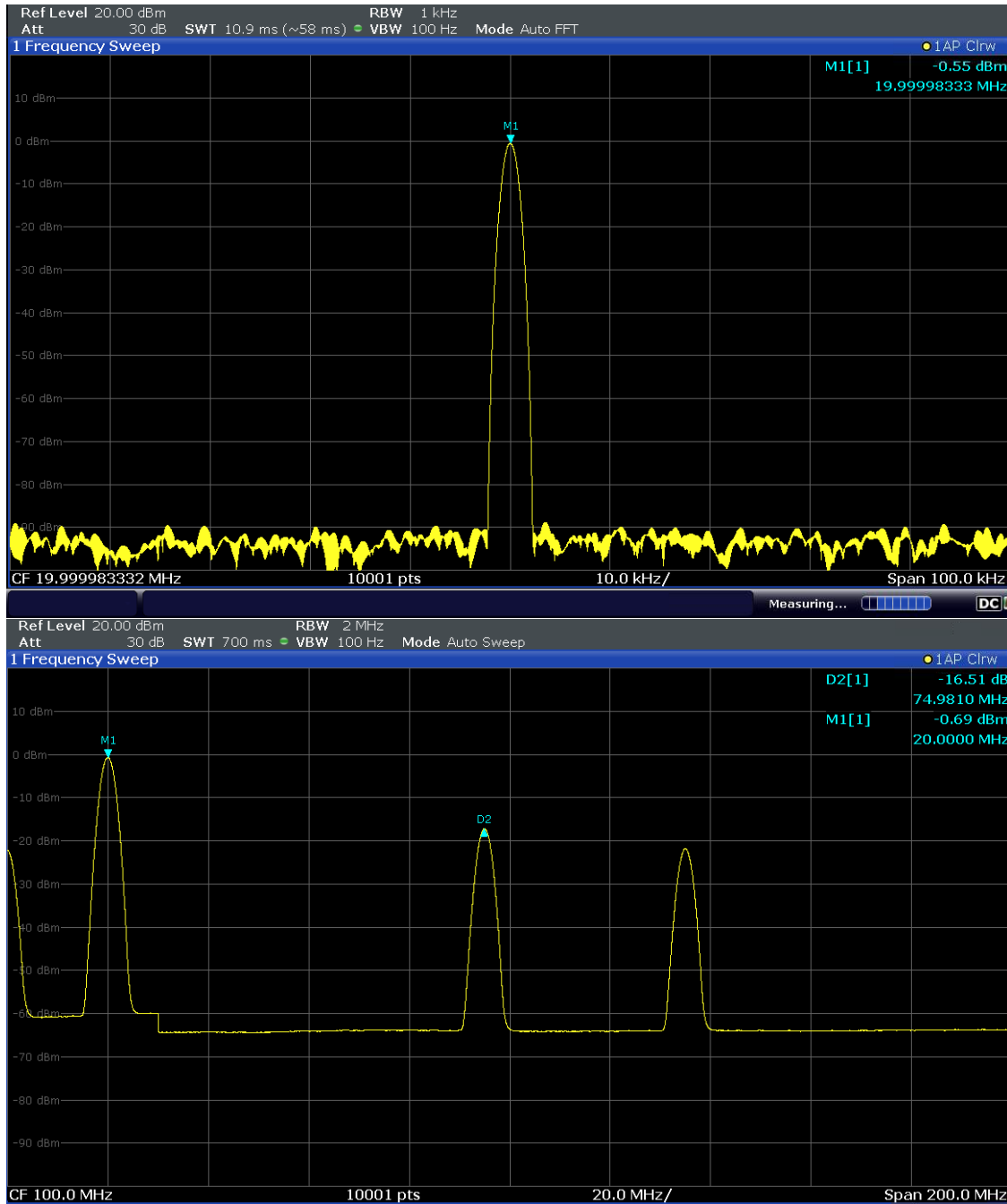
2.2 Check DAC waveform on the oscilloscope

Connect DAC Q1 output to the oscilloscope, expect waveforms like:



2.3 Check DAC frequency response on spectrum analyzer

Connect DAC output to the spectrum analyzer, expect spectrum like:



Compare the measured spectrum with the specified transferfunction in Figure 65 on page 29 in [AD9781 datasheet](#), use the NORMAL curve. Does the DAC function as expected in terms of harmonic distortion?

2.4 Change DAC output frequency

In open loop operation, the DAC output in the LLRF firmware is generated by up-converting the open-loop setpoint using a digital LO DDS module, which we have simulated in the DDS lab. A 32-bit register named `USPAS:LLRF:reg_dds_phase_step` is available through the

EPICS IOC, which has a default value of 746950834, because it is a concatenated value of {phase_step_h, phase_step_l}, or $(182361 \ll 12) \mid 178$.

Example of reading it back using `caget`:

```
$ caget USPAS:LLRF:reg_dds_phase_step_RBV
USPAS:LLRF:reg_dds_phase_step_RBV 746950834
```

Use the `caput` command in terminal to change that register to a value, so that the DAC output signal can be changed from 20.00 MHz to 20.05 MHz

```
$ caput USPAS:LLRF:reg_dds_phase_step <your value>
```

Confirm the signal on the spectrum analyzer.